

Redes de Perceptrons Multi-Camada

Luis Martí

Institute of Computing

Univesidade Federal Fluminense

lmarti@ic.uff.br / <http://lmarti.com>

Sumário

- Introdução
- O elemento de Processamento
- Redes Neurais Artificiais
- Aprendizagem
- Tipos de Redes
- Descrição dos Tipos de Redes

Redes neuronais

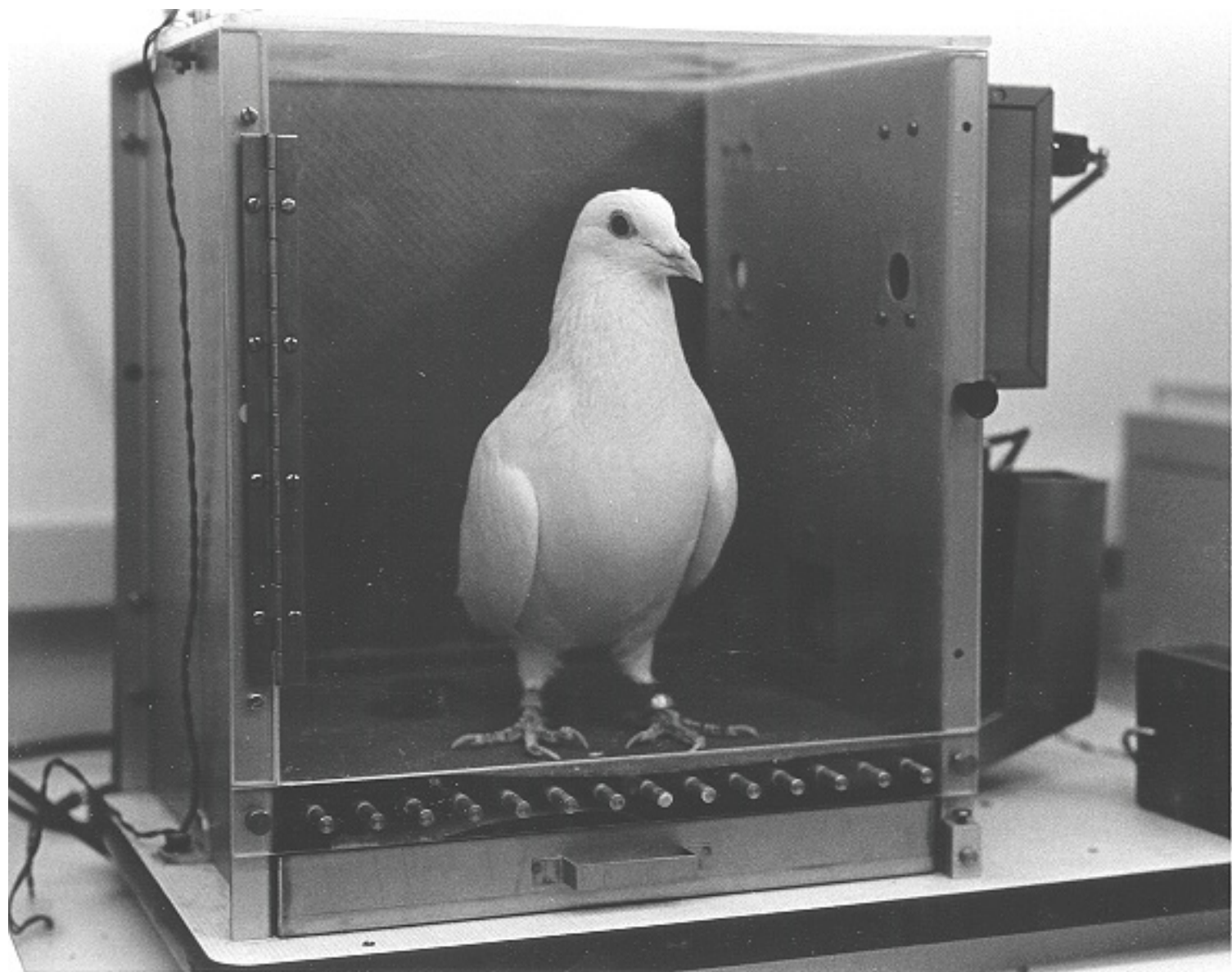
- O que?
- Para que?
- Por que?

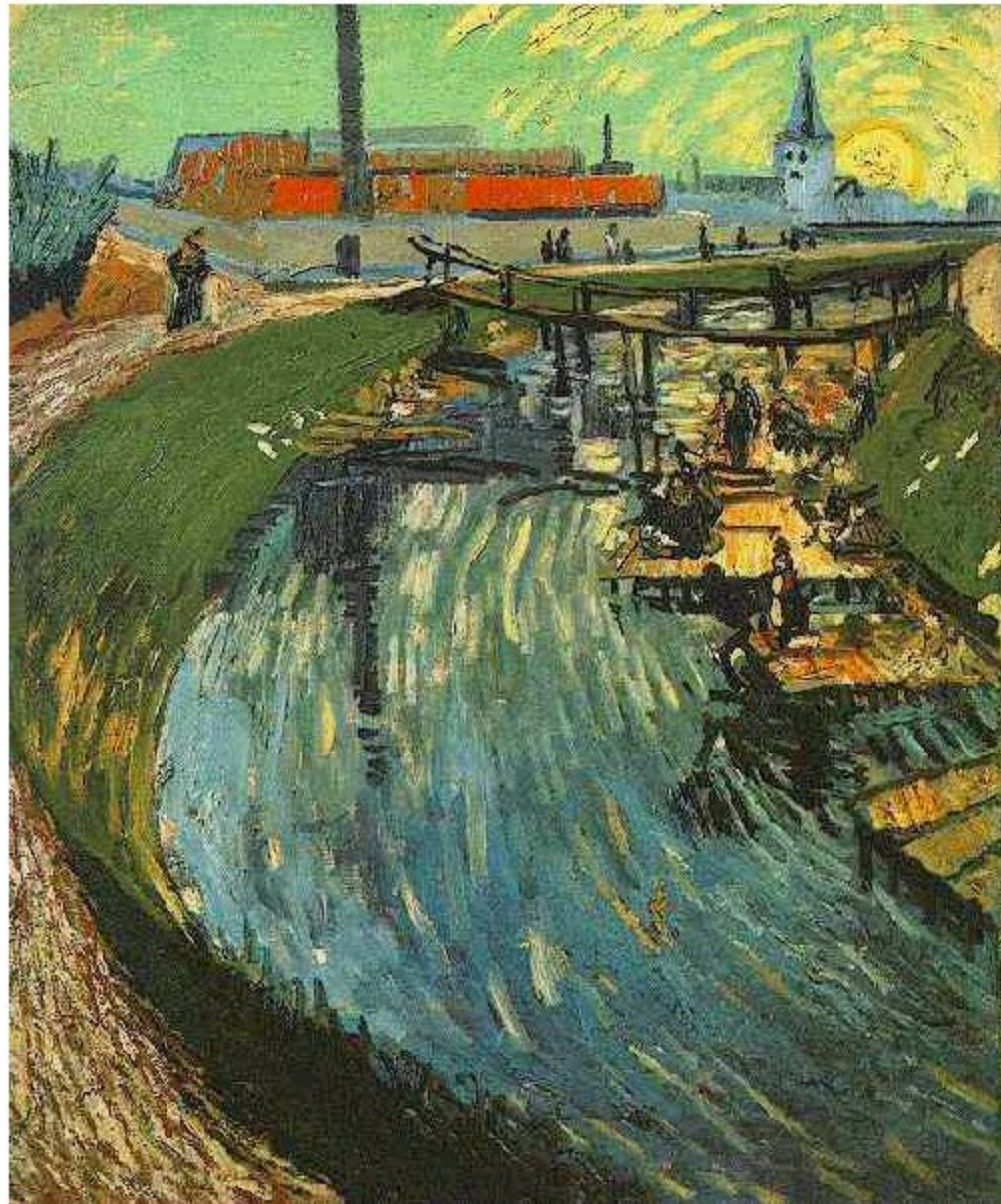
O que são Redes neuronais?

- Modelos do cérebro e do sistema nervoso
- Alto grau de paralelização
 - Processamento de informação muito mais como o cérebro do que como um computador serial
- Aprendizagem
- Princípios muito simples
- Comportamentos muito complexos
- Aplicações
 - Poderosos solucionadores de problemas
 - Modelos biológicos

Redes neuronais biológicas

- Pombos como especialistas em arte (Watanabe *et al.* 1995)
 - Experimento:
 - Pombos em uma caixa de Skinner
 - São apresentadas pinturas de dois diferentes artistas (e.g. Chagall / Van Gogh)
 - Pombos recebem uma recompensa quando apresentados a um particular artista (p. e. Van Gogh)





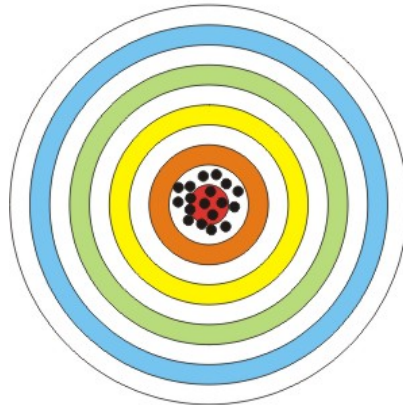


Redes neuronais biológicas

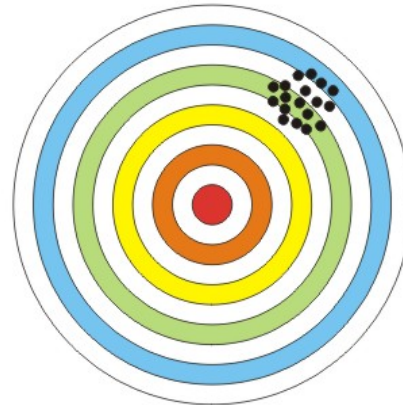
- Pombos foram capazes de discriminar entre Van Gogh e Chagall com acurácia de 95% (quando foram apresentados a pinturas com as quais haviam sido treinados)
 - Para pinturas dos mesmos artistas que ainda não haviam sido vistas pelos pombos a discriminação ficou em 85%
- Pombos não memorizam simplesmente as pinturas
 - Eles podem extrair e reconhecer padrões (o 'estilo')
 - Eles generalizam a partir do que já viram para fazer previsões
- Nisto é que as Redes neuronais (biológicas ou artificiais) são boas (ao contrário dos computadores convencionais)

Acurácia e precisão

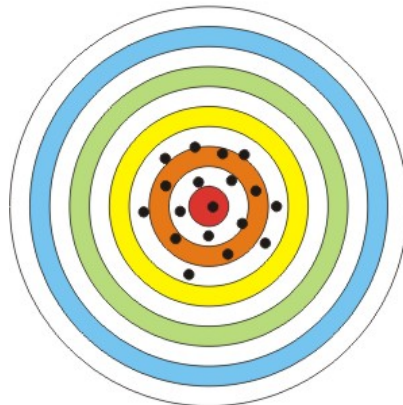
Alta acurácia
Alta precisão



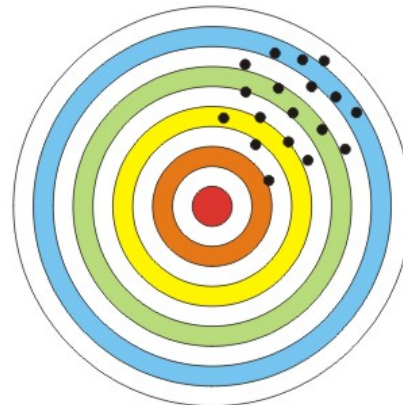
Baixa acurácia
Alta precisão



Alta acurácia
Baixa precisão



Baixa acurácia
Baixa precisão



Para que?

- Para resolver um problema no qual exista incerteza sobre um dado fenômeno.
- O usuário levanta informações que julga poder ajudar na solução do problema ou redução da incerteza.

Exemplo

Atr. 1	Atr. 2	Atr. 3	Atr. 4	Atr. 5
71.943	46.163	15.195	18.600	2.359
73.097	46.789	15.413	18.600	2.363
72.513	50.634	15.936	19.000	2.358
77.277	52.615	16.107	19.200	2.356
81.325	54.349	18.507	20.800	2.346
82.457	53.759	20.661	21.300	2.348
81.627	50.253	20.302	19.300	2.372
81.851	41.394	20.257	19.500	2.371
80.807	40.650	19.834	22.100	2.323
80.368	41.439	19.318	22.400	2.305

Interpretação da incerteza

- Pode-se imaginar que cada atributo seja uma coordenada do ponto representativo da amostra, ou instância, em um hiper-espaço cuja dimensão é o número de atributos.

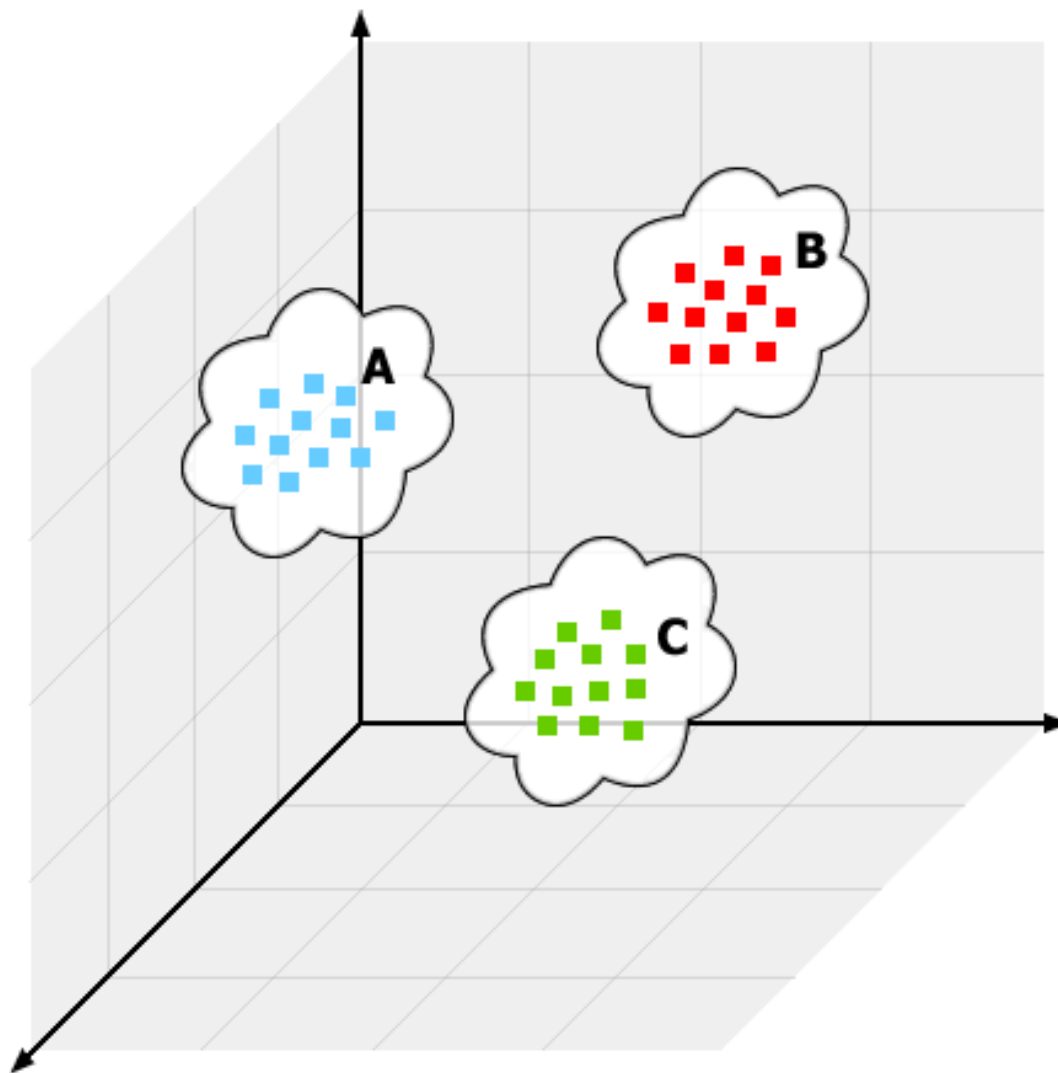
Tipos de problema

- Classificação
- Regressão
- Análise de Agrupamentos

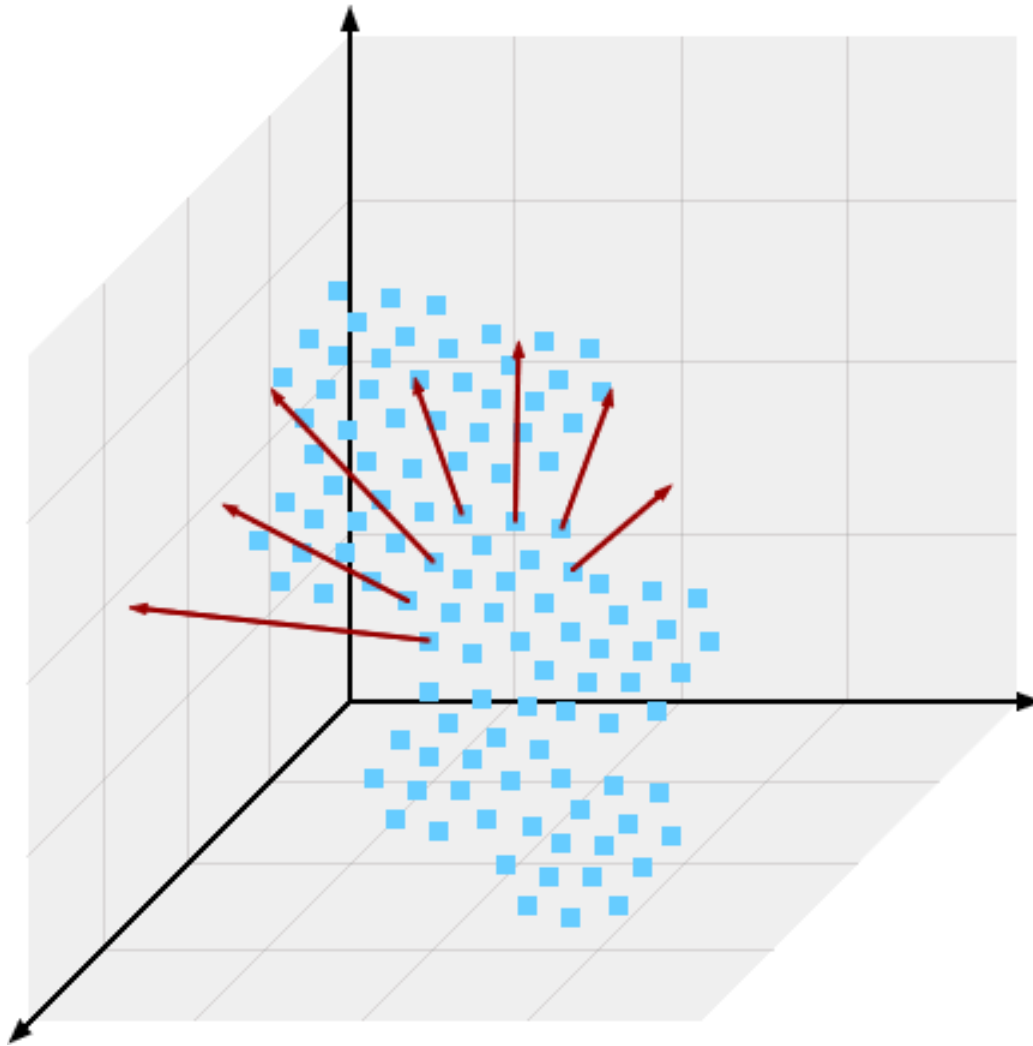
Classificação e Regressão

- Classificação é a atribuição de casos ou instâncias de dados a uma ou mais possíveis classes.
 - Em Redes neurais freqüentemente existe um elemento de processamento por classe.
- Regressão é a estimativa do valor de uma variável baseada em exemplos.

Classificação



Regressão

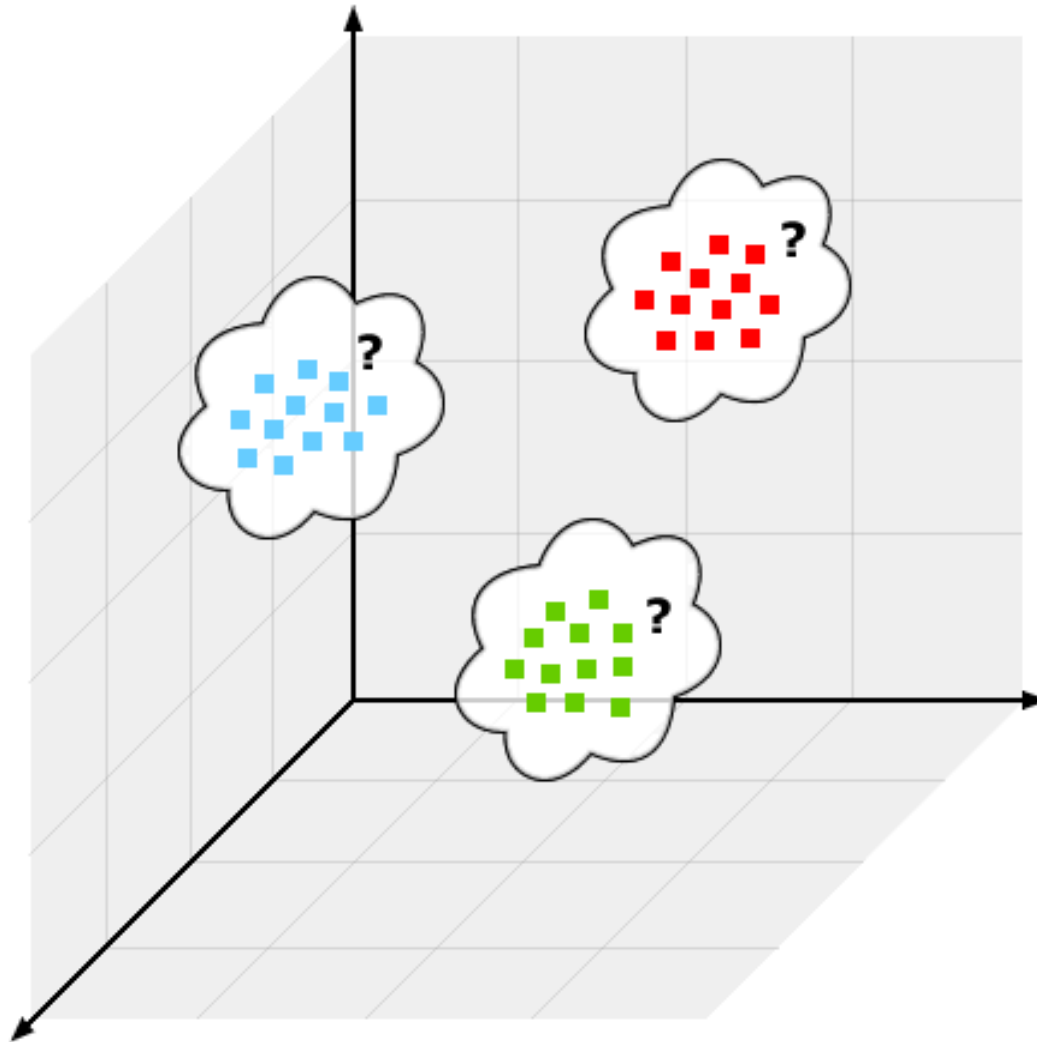


Análise de Agrupamentos

O objetivo é:

- agrupar objetos físicos ou abstratos em classes de objetos *similares*, chamados de agrupamentos (*clusters*).

Análise de Agrupamentos



Por que?

- Porque há necessidade de uma metodologia para balizar as tentativas de obtenção de uma solução aproximada.
- Existem outras metodologias:
 - Estatística
 - Lógica Fuzzy
 - Médias móveis
 - KNN

A metodologia de Redes neuronais é intelectualmente fascinante e dá bons resultados.

Taxonomia de Redes neuronais

As Redes neuronais podem ser classificadas em:

- Redes com pesos fixos
- Redes supervisionadas
- Redes não supervisionadas

Redes neuronais com pesos fixos

- Redes com pesos fixos são aquelas nas quais os pesos das sinapses (conexões entre elementos de processamento) são fixos e armazenados.
 - **modelos de Hopfield.**

Redes neuronais Supervisionadas

- Redes Supervisionadas são aquelas nas quais os padrões de treinamento devem ser fornecidos em pares do tipo
 - **<Entrada, Saída correspondente>**
- São supervisionados a maioria dos paradigmas de Redes neuronais:
 - Perceptrons multi-camada com retro-propagação do erro.
 - Redes de base radial.
 - *Probabilistic neural networks* (PNN)
 - *General regression neural networks* (GRNN)
 - *Convolutional neural networks*
 - Etc.

Redes Neurais Não Supervisionadas

- São aquelas nas quais os padrões de treinamento contém apenas a Entrada.
- São Não Supervisionados os paradigmas de
 - ART
 - Kohonen
 - Contra propagação
 - Etc.

Aplicação das Redes neuronais Não Supervisionadas

- A aplicação característica das Redes neuronais Não Supervisionadas é a Análise de Agrupamentos.
- Pode-se dizer que as Redes neuronais fazem o mapeamento de
 $R^n \Rightarrow (Z *)^2$ ou $R^n \Rightarrow (Z +)^2 = N^2$

Introdução às Redes Neuronais Artificiais

Introdução às RNAs

- Definições
- Regras
- Modelos de neurônios
- Aplicações

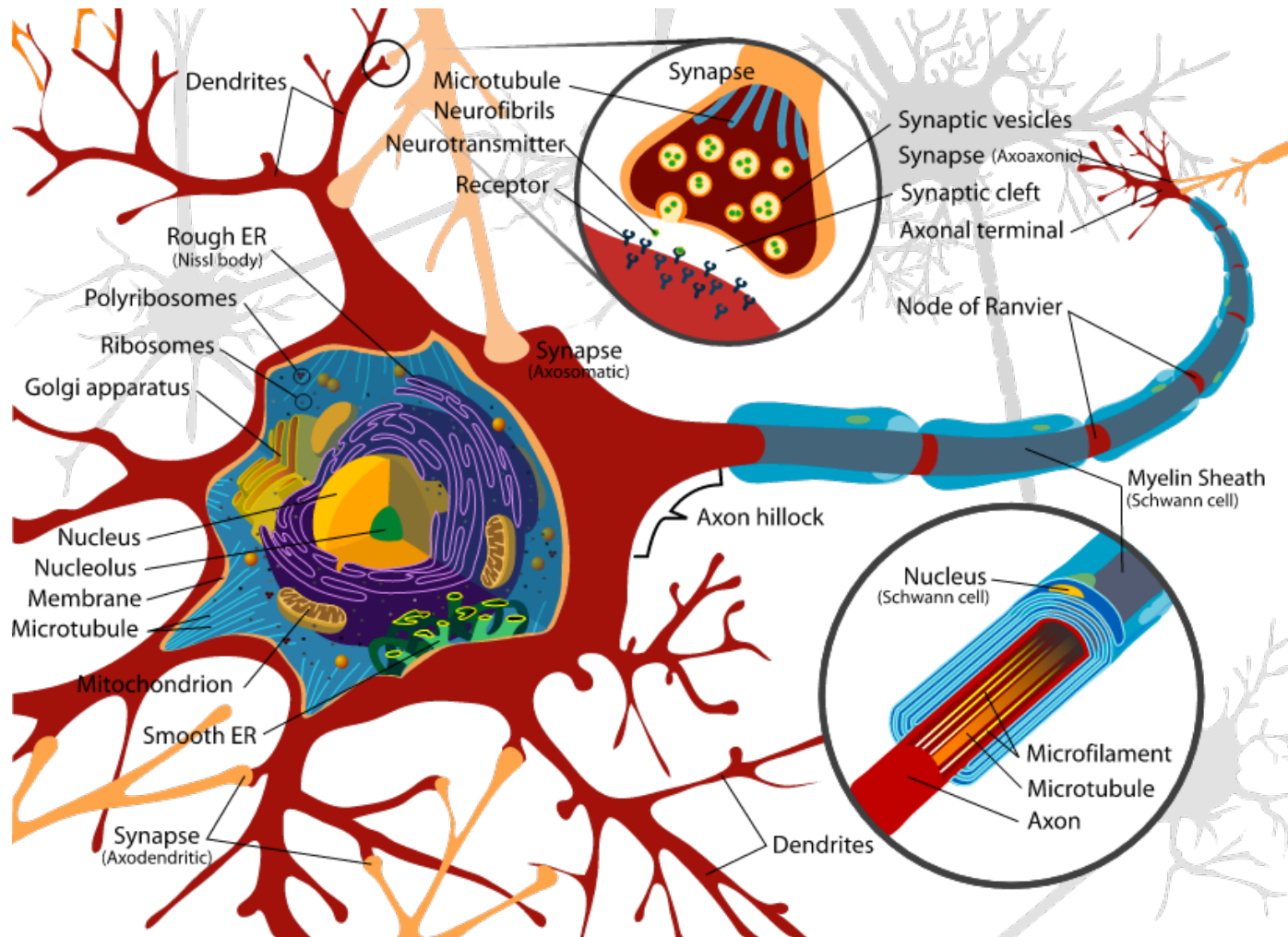
Definição

Uma **rede neuronal artificial** é uma **construção matemática simplificada inspirada** no **modelo biológico do sistema nervoso** dos animais.

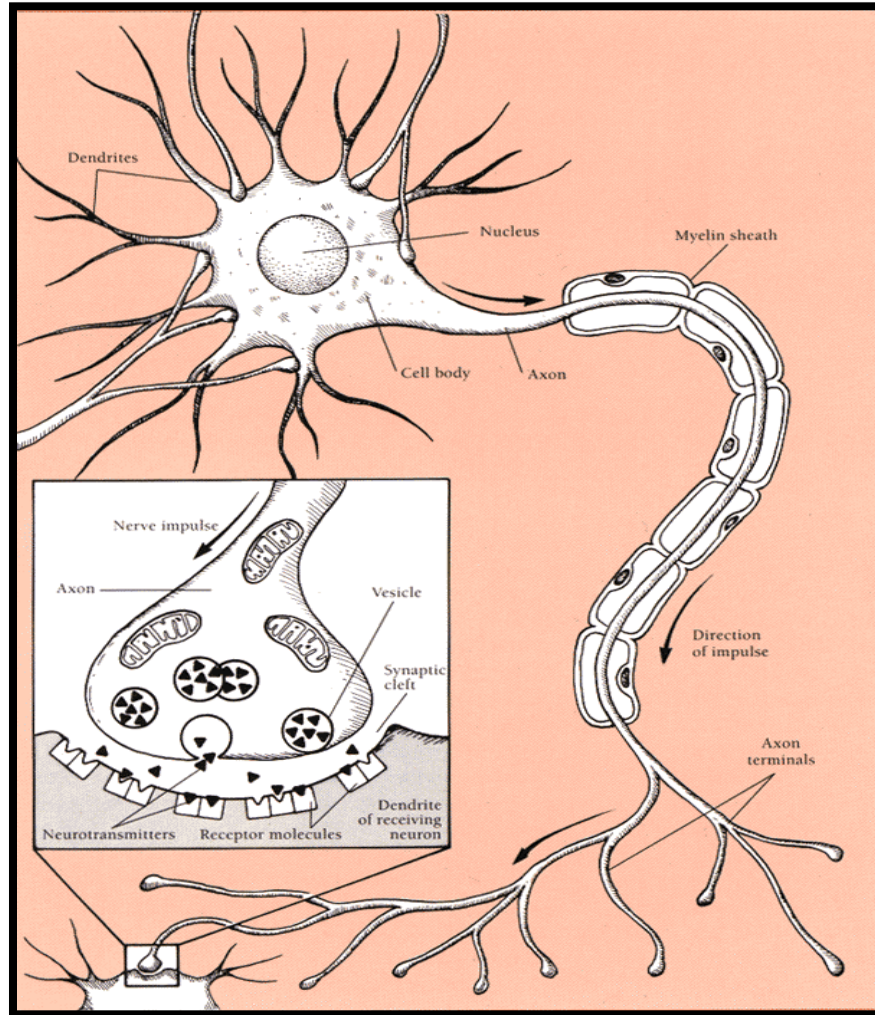
Neurônios

- O principal componente do sistema nervoso dos animais é uma célula denominada neurônio, que funciona como um **elemento de processamento** ou **processador**
- Seu aspecto esquemático é mostrado na figura que se segue

Neurônio natural



Neurônio natural



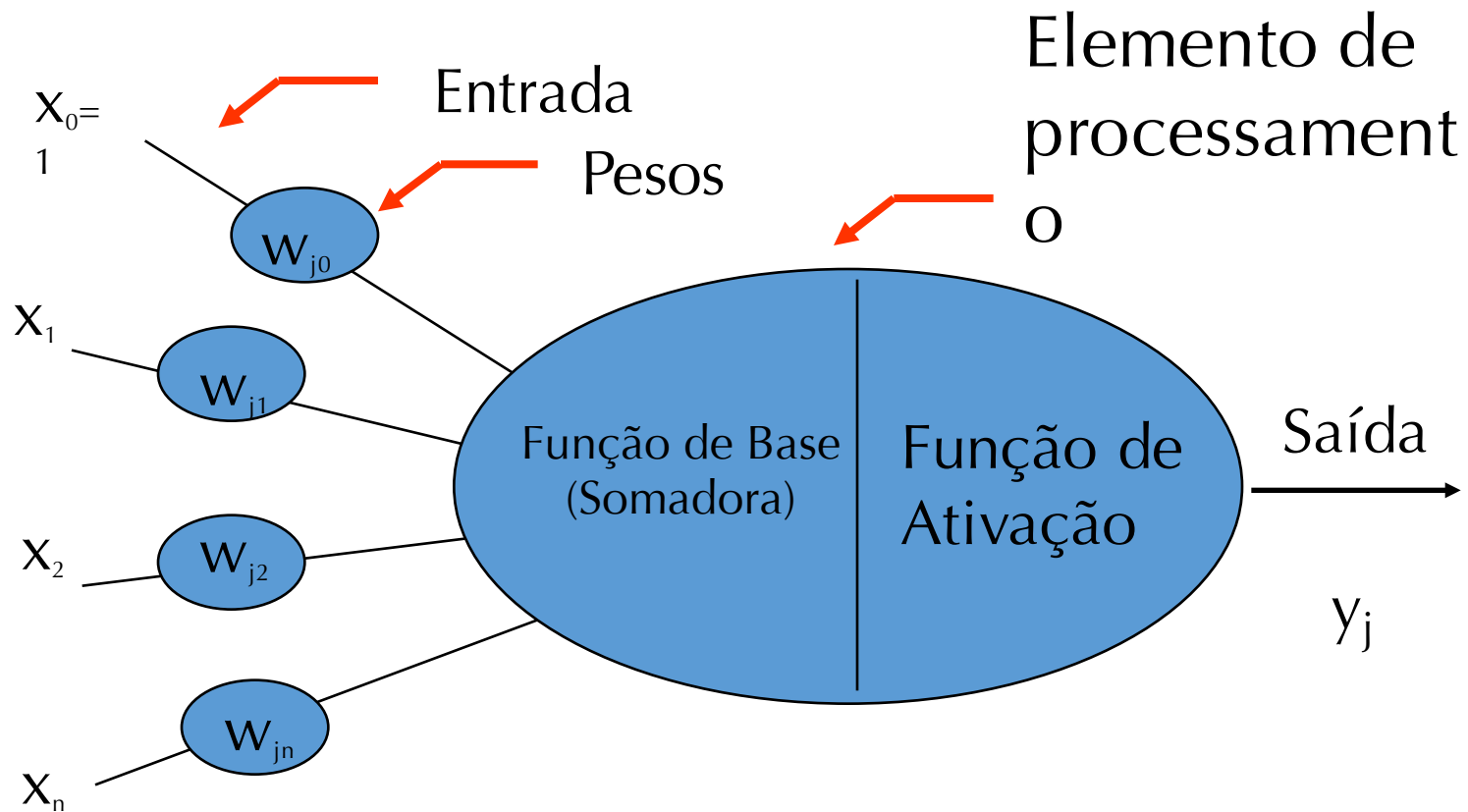
Neurônios artificiais

- Elemento de processamento (EP) ou unidade de processamento ou neurônio artificial é composto de uma **função de entrada** ou **função somadora** e de uma **função de saída** ou **função de ativação** ou **função de patamar**.
- Cada elemento de processamento pode receber um ou mais dados de entrada sendo cada um deles proveniente do meio ambiente ou de outro neurônio.
- Em uma unidade de tempo cada conexão só recebe um dado de entrada
- Um elemento de processamento só tem uma saída
- A saída pode ser direcionada, em paralelo, a diversos neurônios

O Elemento de Processamento

- Função de Base
- Função de Ativação
- Conexão entre neurônios

Modelo de Elemento de Processamento



Conexões entre neurônios

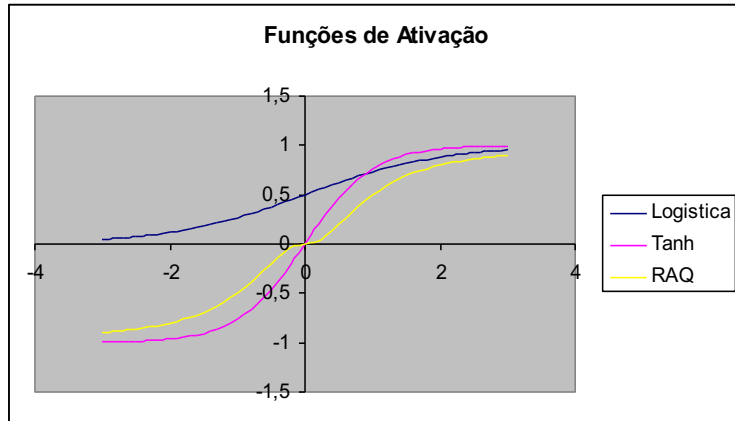
- As conexões entre neurônios são ponderadas
- Uma conexão virtual especial chamada de conexão ou entrada de polarização serve para implementar o conceito de valor de patamar
- Em um neurônio com n conexões de entrada o valor da entrada na sinapse de ordem i é x_i
- O valor da entrada de polarização é sempre $x_0=1$

Função de Base Somadora

- A função somadora pode ser apresentada como:

$$I = \text{net}(x) = \sum_{i=0}^n w_i x_i$$

Função de Ativação



- Transformam números reais em números entre 0 e 1 ou entre -1 e 1.
- Mapeiam o domínio da ativação do neurônio no domínio de saída
- Funções usuais:
 - Linear, rampa, degrau, sigmóides (logística ou tangente hiperbólica)

Funções de Base Global

- Definidas como funções da distancia do vetor de padrões a um hiperplano
 - Sua base é global pois assumem valores em todo o domínio de definição do problema
 - A função a ser aproximada se torna uma combinação de sigmoidais, que sendo definidas em todo o espaço de medidas exigem muitas iterações até chegar a uma combinação adequada

Funções de Base Local

- Definidas na vizinhança de um elemento de processamento assumindo valores negligíveis fora da vizinhança desse elemento, podendo ser de dois tipos:
 - Baseadas em estimativa das funções de densidade de probabilidade
 - Baseadas em aproximação de funções iterativas

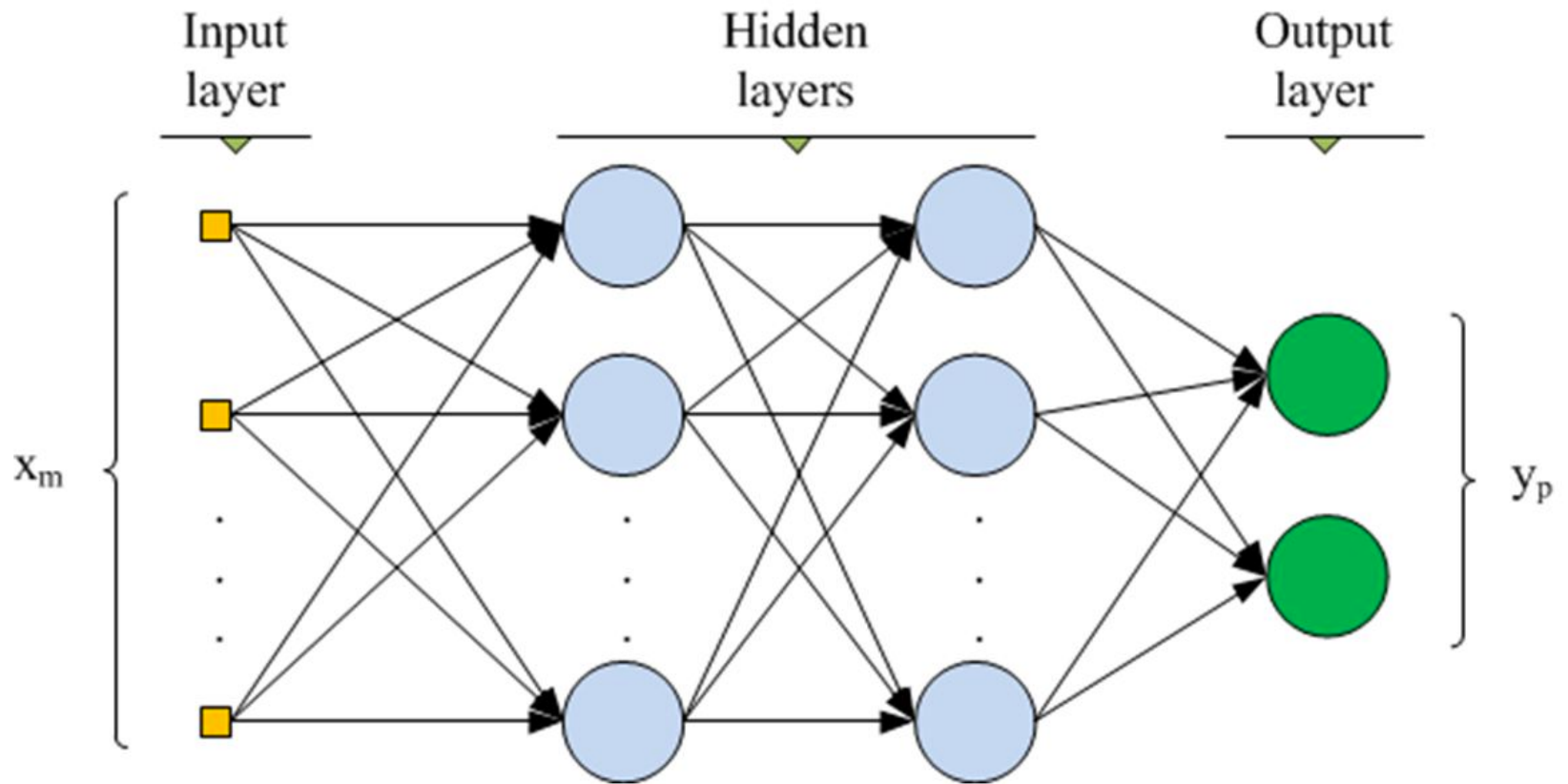
Redes Neurais Artificiais

- Conceito
- Representação do Conhecimento e Centralização de Controle
- Operações

Rede Neural Artificial

- Uma Rede Neural Artificial consiste de diversos elementos de processamento interconectados.
- Esses elementos usualmente são organizados em grupos denominados **camadas**.
- Redes neuronais constituem-se em sequências de camadas com conexões entre elas (completas ou aleatórias).
- Destacam-se duas camadas, de contato com o exterior: a **camada de entrada** de dados e a **camada de saída**.
- Todas as demais camadas porventura existentes são chamadas de **camadas ocultas**.

Exemplo de RNA



Arquitetura das redes neuronais

Topologia

- Camada única (Perceptron)
- Multicamadas (camadas ocultas)

Tipo de aprendizado

- Supervisionado
- Não supervisionado

Classificação dinâmica

- Retroalimentação
- Feed forward

Redes neuronais

Camadas

- Camada de entrada
- Camadas ocultas
- Camada de saída

Função de base

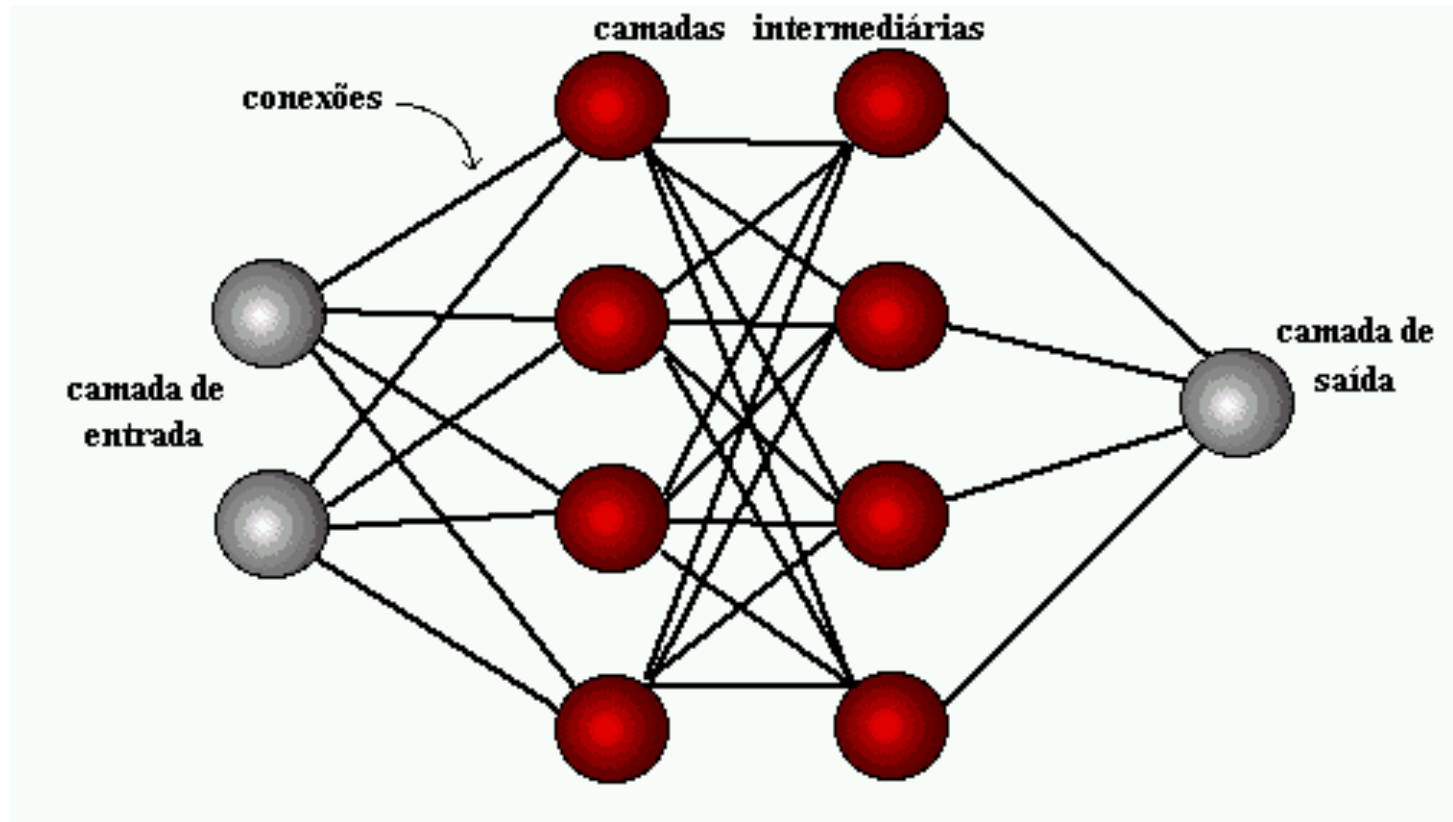
- Combinação das entradas

Função de ativação

- Saída

Aprendizado

Organização em camadas



Organização em camadas

As camadas são classificadas em três grupos:

- Camada de entrada:
 - **Padrões são apresentados à rede.**
- Camadas intermediárias ou escondidas:
 - **Concentram a maior parte do processamento através das conexões ponderadas.**
 - Podem ser consideradas como extratoras de características
- Camada de saída:
 - Onde o resultado final é computado e apresentado de volta.

Operação das Redes neuronais

A operação das Redes neuronais compreende duas fases :

1. Aquisição do conhecimento ou treinamento ou aprendizagem.
2. Recuperação (uso).

Aquisição de conhecimento

- O **treinamento** consiste na **adaptação** ou **modificação** dos pesos das conexões em resposta a estímulos apresentados à camada de entrada e, opcionalmente, à camada de saída.
- Um estímulo apresentado à camada de saída corresponde à resposta desejada a um estímulo apresentado à camada de entrada.
- Quando isto acontece ocorre a aprendizagem supervisionada.
- Caso não seja fornecida nenhuma saída ocorre a aprendizagem não supervisionada.

Recuperação

- A recuperação ou utilização é a obtenção da resposta gerada pela Rede Neuronal, em sua camada de saída em reação a um estímulo apresentado à camada de entrada.

Aprendizagem

- Como já vimos a aprendizagem pode ser classificada em:
 - Aprendizagem Não supervisionada
 - Aprendizagem Supervisionada
 - Aprendizagem por Reforço

Aprendizagem não supervisionada

- Na presença apenas de estímulos de entrada a rede se organiza internamente.
- Cada EP responde de maneira mais intensa a um grupo diferente de estímulos.
- Estes grupos dos conjuntos de estímulos representam distintos conceitos do mundo real.
- Podem ser usados os seguintes tipos de aprendizagem:
 - Hebbian (devida a “Hebb”)
 - Hopfield
 - Aprendizagem Competitiva

Regra de Hebb

“Quando um axônio de uma **célula A** está próximo o suficiente de **excitar** a **célula B** e repetidamente ou persistentemente participa da ativação desta, um processo de crescimento ou mudança metabólica ocorre em uma ou ambas as células, de tal forma que a **eficiência** de **A** em **ativar B** é **aumentada**”

Portanto, a cada apresentação do padrão a saída fica mais reforçada

Regra de Hebb

Em termos práticos:

- Se dois neurônios em cada lado de uma sinapse (conexão) **são ativados simultaneamente** (sincronamente), então a “força” daquela **sinapse deve ser aumentada**
- Se dois neurônios em cada lado de uma sinapse **são ativados assincronamente**, então aquela **sinapse deve ser enfraquecida**

Aprendizagem de Hopfield

- A aprendizagem de Hopfield baseia-se no sistema olfativo de uma lesma de jardim modelado em um sistema computacional de elementos de processamento interconectados buscando a energia mínima para o sistema.
- O funcionamento dos neurônios é uma operação de patamar e a memória consiste em informação armazenada nas conexões entre neurônios.

Aprendizagem competitiva

- Regra de aprendizagem na qual os elementos de processamento competem para responder a um dado estímulo
- O vencedor adapta-se para tornar-se ainda mais próximo ao estímulo

Aprendizagem competitiva

- A frase que caracteriza este tipo de aprendizagem é **“o vencedor leva tudo”**.
- Para que isto ocorra as unidades de saída são completamente conectadas umas às outras, sendo que os pesos destas conexões são todos negativos
- Com estes pesos cada elemento procura inibir a ativação de todos os demais elementos
- Este tipo de conexão leva a tendências de vitória irresistíveis

Aprendizagem supervisionada

- Para cada estímulo a rede se **adapta** para **gerar** uma saída próxima do estímulo de saída.
- Pode ser dos tipos:
 - Regra Delta
 - Gradiente Descendente
 - Delta Barra Delta
 - Delta Barra Delta Estendida

Regra Delta

- Algoritmo que fornece **convergência** para o único conjunto de pesos que dá o **menor erro quadrático médio** entre as saídas desejadas e obtidas para o conjunto do exemplo.

Regra Delta

- Baseia-se na modificação dos pesos das conexões para reduzir a diferença (delta, Δ) entre a saída desejada e a saída real de um elemento de processamento
- As modificações minimizam o erro médio quadrático da Rede

Regra Delta

- O erro delta da camada de saída é transformado pela derivada da função de transferência e é usado na camada anterior da Rede para ajustar o peso das conexões de entrada
- O erro é propagado para trás para as camadas anteriores, uma de cada vez, até atingir a camada de entrada

Regra do Gradiente Descendente

- Semelhante à Regra Delta pois também usa a derivada da função de transferência para modificar o erro delta
- A diferença é o uso de uma constante de proporcionalidade da taxa de aprendizagem juntada ao fator final de modificação
- Converge mais lentamente que a Regra Delta

Estratégia de Aprendizagem

- A regra de aprendizagem especifica a maneira como os pesos se adaptam em resposta aos exemplos de treinamento (estímulos de entrada)
- Parâmetros que governam a regra de aprendizagem podem variar com o tempo, à medida que a aprendizagem progride
- O controle dessa variação de parâmetros é chamado de Estratégia de Aprendizagem ("learning schedule")

Tipos de Redes

- Propagação dos estímulos
- Listagem dos tipos

Propagação dos estímulos

- Estímulos apresentados à camada de entrada podem se propagar aos elementos das demais camadas, com alimentação para diante em uma "feedforward network"
- Cada elemento que receba um estímulo o propaga usando suas funções de soma e de propagação
- Em alguns tipos de redes existe, também, propagação para trás ou retro alimentação caracterizando "backpropagation network"

Tipos de Redes ou Paradigmas de Redes

- Hopfield
- Perceptron
- Retro Propagação
- Boltzmann
- Contra propagação
- Regressão Geral
- LVQ
- STN
- Base Radial
- PNN

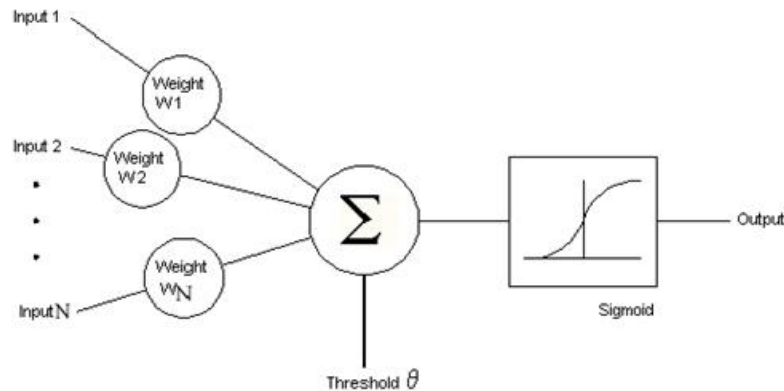
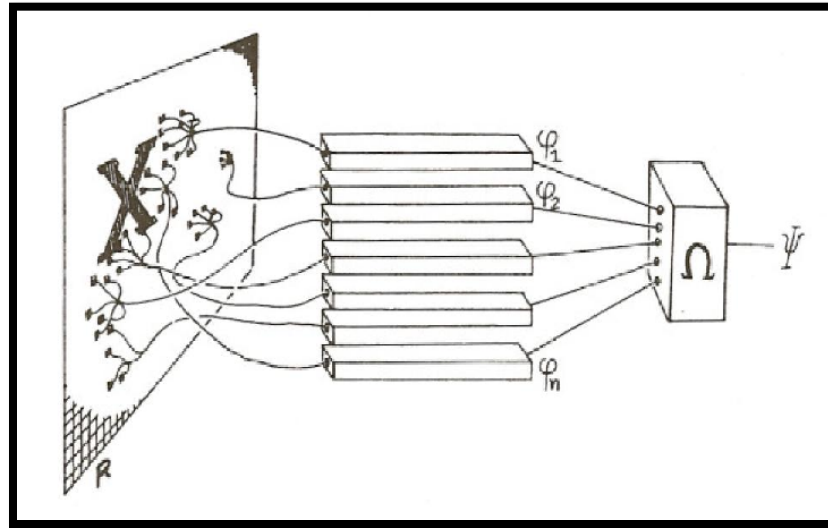
Redes neuronais Supervisionadas

Perceptron

- Proposto por Rosenblatt (1959) para reconhecimento de letras maiúsculas do alfabeto.
- É uma rede direta consistindo de unidades binárias, que aprendem a classificar padrões através de aprendizado supervisionado
- Introduz formalmente uma lei de treinamento

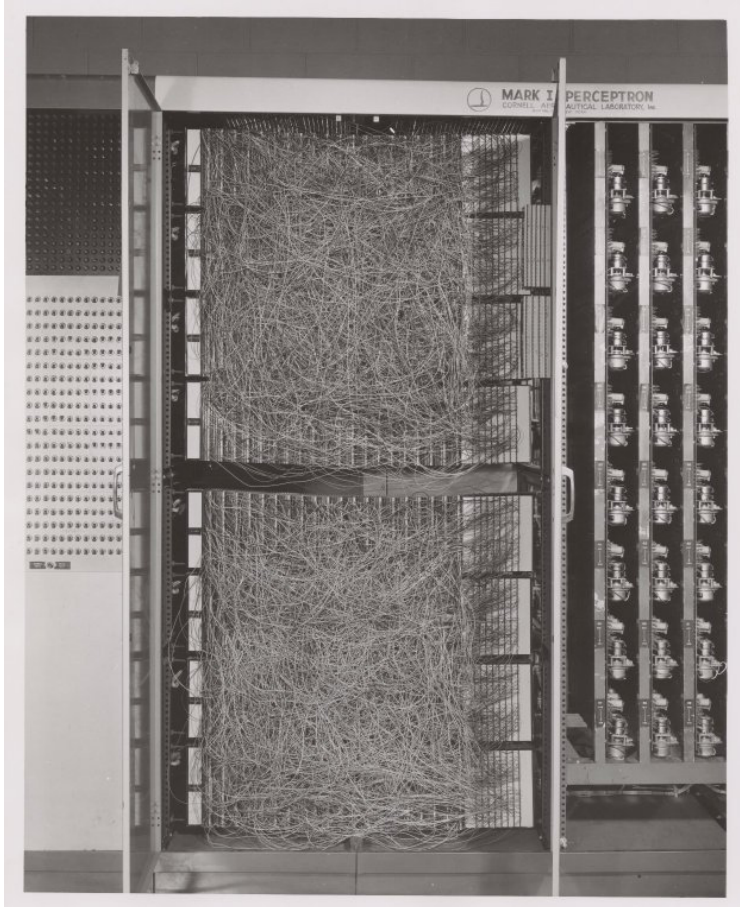
Modela o neurônio fazendo a soma ponderada de suas entradas e enviando o resultado 1 se a soma for maior do que algum resultado inicial ajustável (caso contrário, ele envia 0)

○ Perceptron



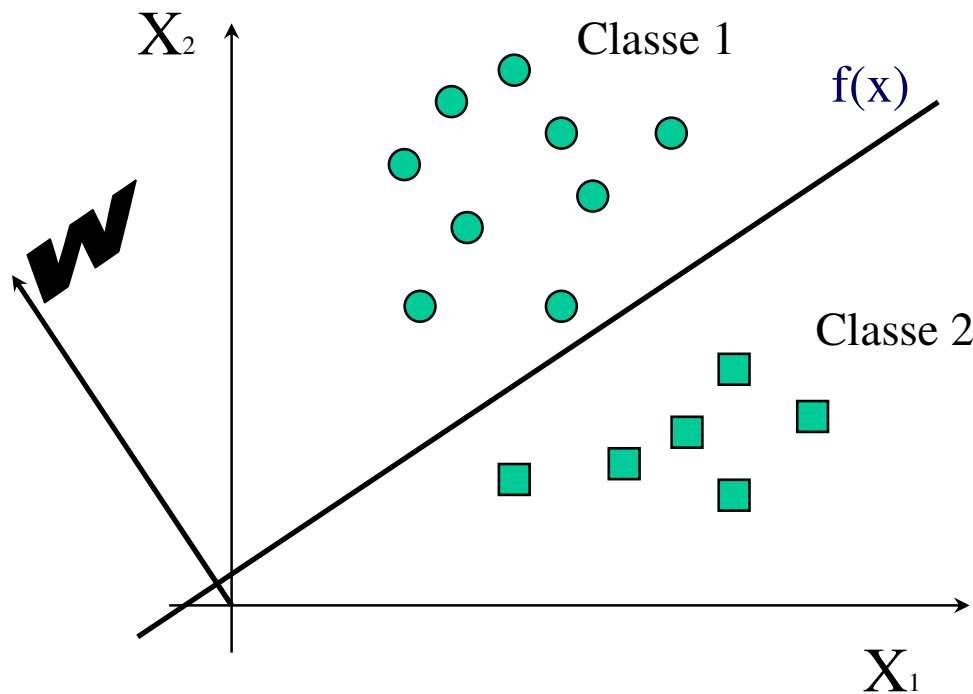
$$y = f\left(\sum_{i=1}^n w_i x_i - \theta\right)$$

Mark I Perceptron



- Uma câmara produzia imagem de 400 pixels (20x20)
- Quadro de conexões seleciona que entradas são passadas para os perceptrons.
- Potenciômetros como pesos.
- Motores elétricos modificavam os pesos.

Uma Visão Matemática do Perceptron



$$f(x) = \sum w_i \cdot x_i - \theta$$

$$f(x) = (|W| \cdot |X| \cos \Phi) - \theta$$

Considere o ponto onde

$$f(x) = 0:$$

$$w_1 \cdot x_1 + w_2 \cdot x_2 - \theta = 0$$



$$x_2 = -w_1/w_2 \cdot x_1 + \theta/w_2$$

$$(y = m \cdot x + c)$$

Perceptron: problemas

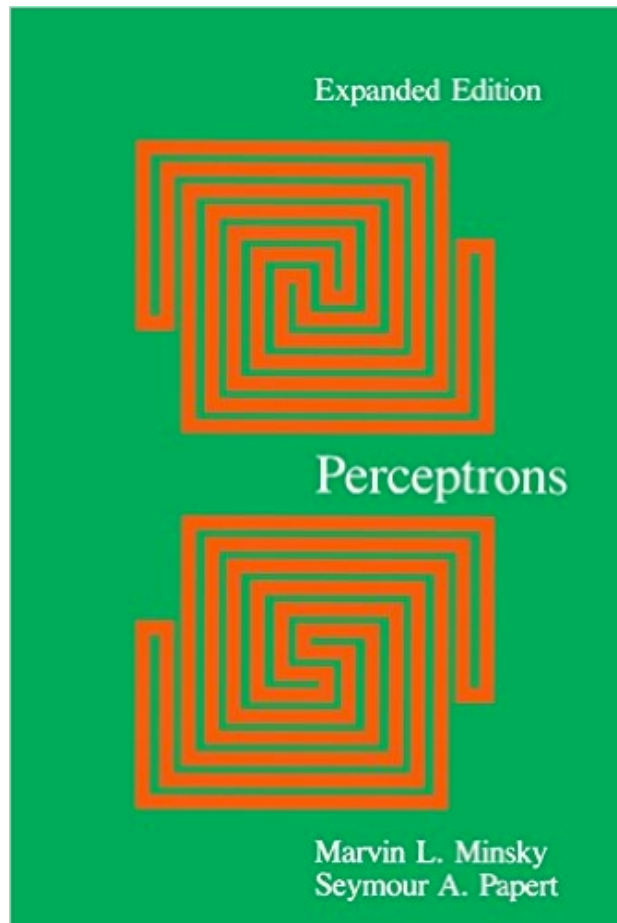
Rosenblatt (1962) provou que:

*Uma rede Perceptron é capaz de **Aprender** tudo que puder **Representar***

Representação refere-se à habilidade do sistema neural de representar **(simular)** uma função específica.

Aprendizado refere-se à existência de um procedimento sistemático de aquisição de conhecimento **(ajuste dos pesos)**, de forma a produzir a função desejada

Perceptron: Problema



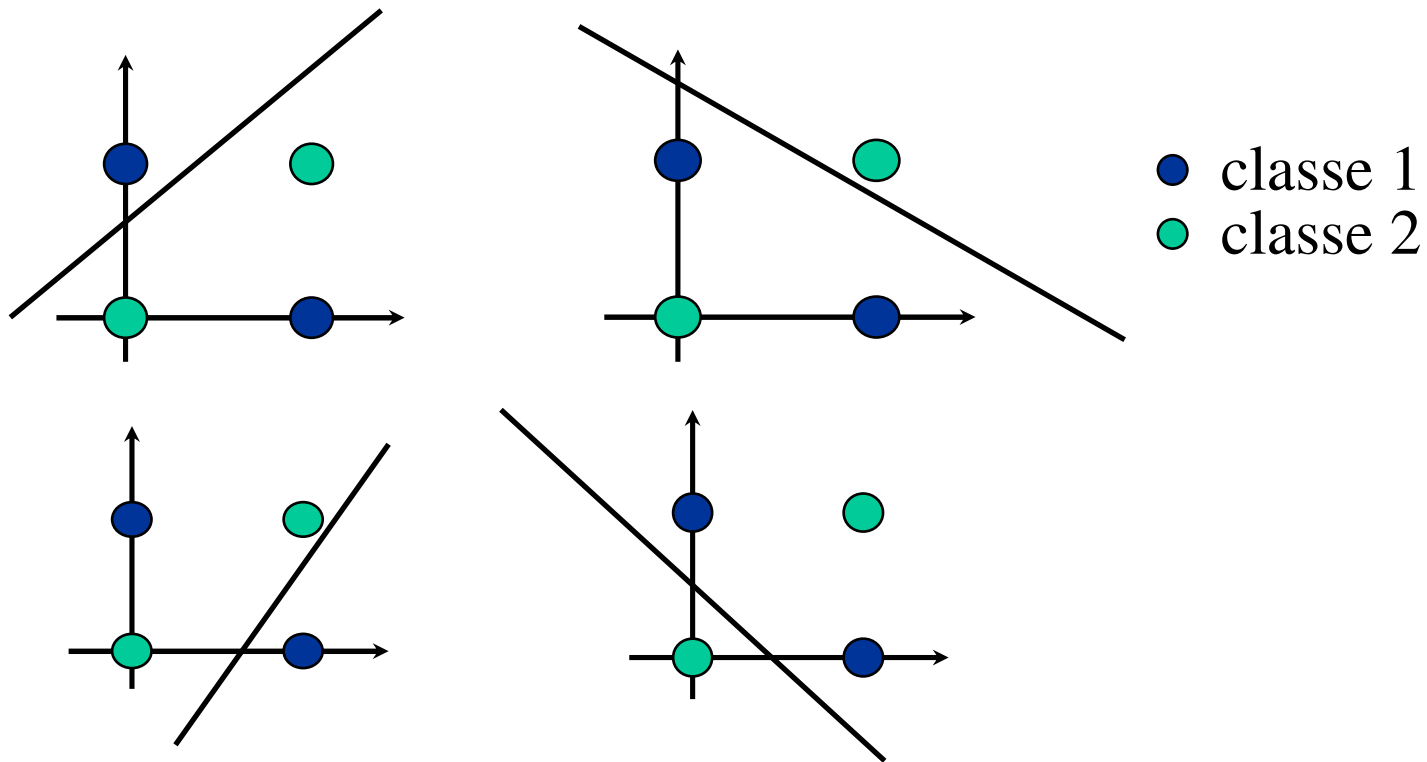
- Minsky & Papert provaram (*Perceptrons* 1969) que existem séries restrições sobre o que as redes Perceptron são capazes de **representar**.
- Por exemplo, as redes Perceptron **não** são capazes de **representar** a função **OU-Exclusivo**.

Ver exemplo

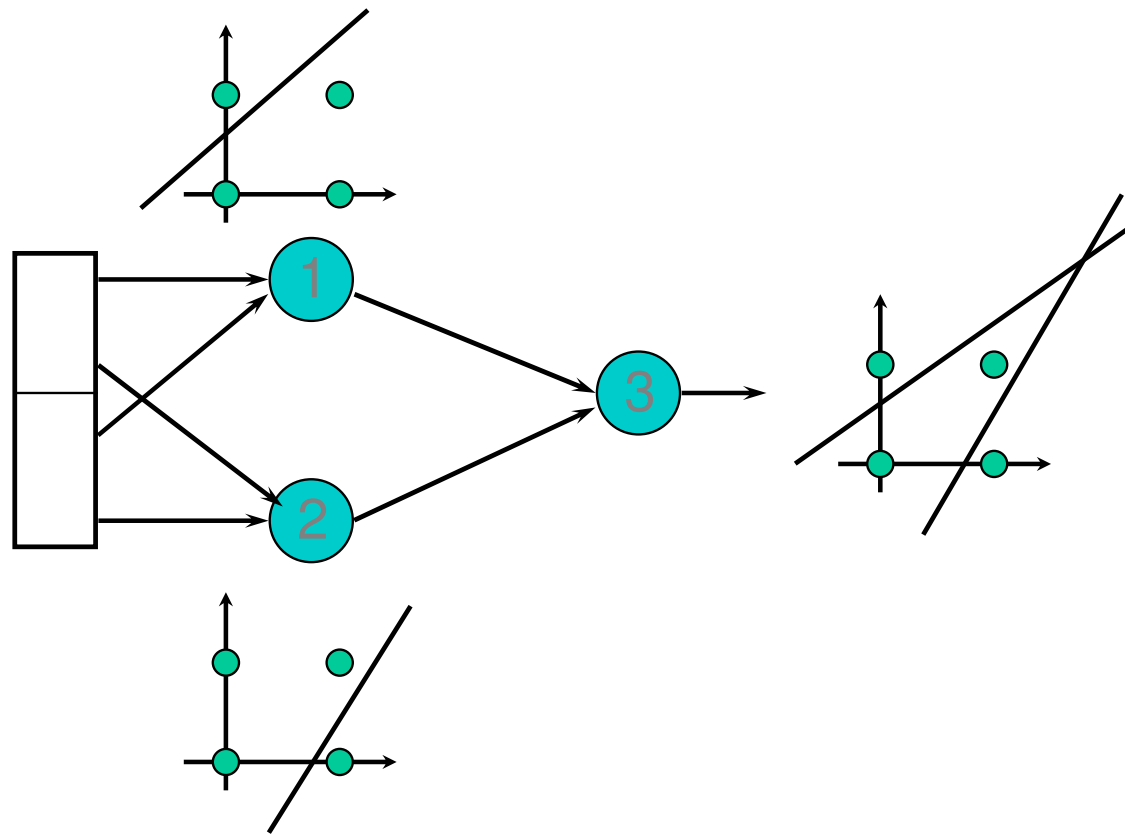
<http://nbviewer.jupyter.org/github/Imarti/machine-learning/blob/master/Understanding%20the%20Perceptron.ipynb>

Perceptron: Problema

Não funciona com classes não-linearmente separáveis como OU-exclusivo.



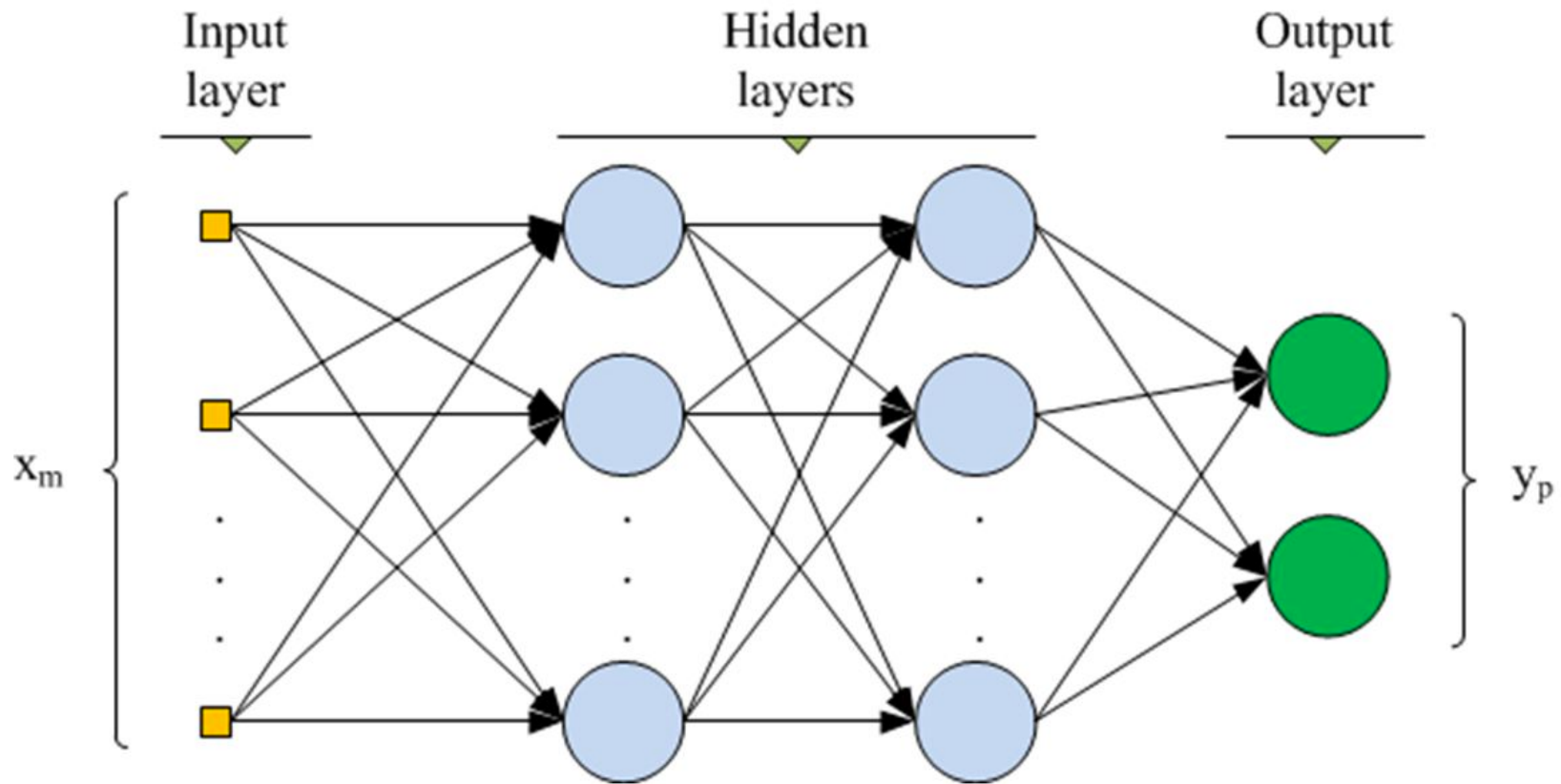
Solução XOR com camadas



Perceptrons multi-camadas

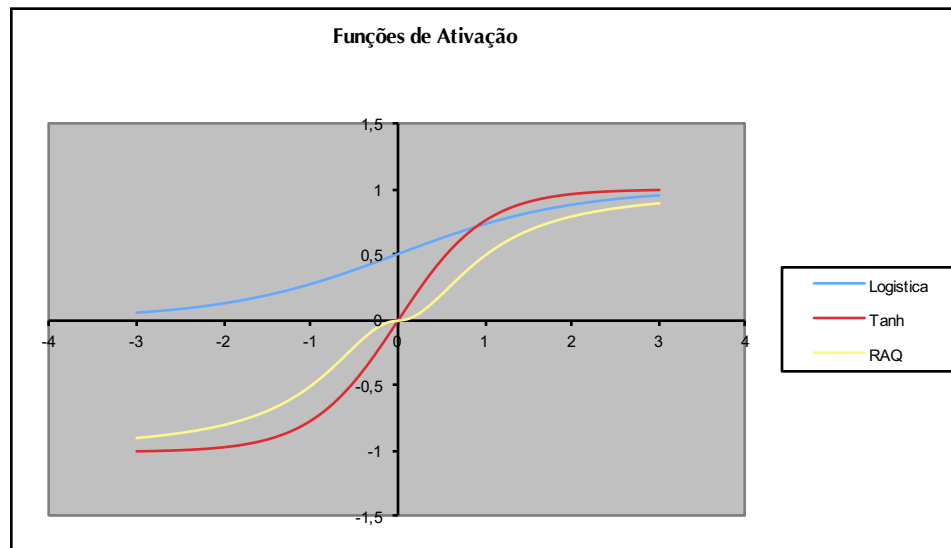
- Perceptrons multi-camadas ou Multi-layer perceptrons (MLP) começaram a ser desenvolvidos para ocupar o espaço deixado pelas limitações dos perceptrons.
- Werbos (1974) criou o **algoritmo de backpropagation** ou **retro-propagação** que permitiu o uso de uma rede neuronal de três (ou mais) camadas.

Rede de Perceptrons Multi-camada



Perceptron extendido

- Substitui a função de ativação por limiar por outras funções contínuas.
- ...e deriváveis. <- *Fundamental para aplicar o algoritmo que vamos a ver.*



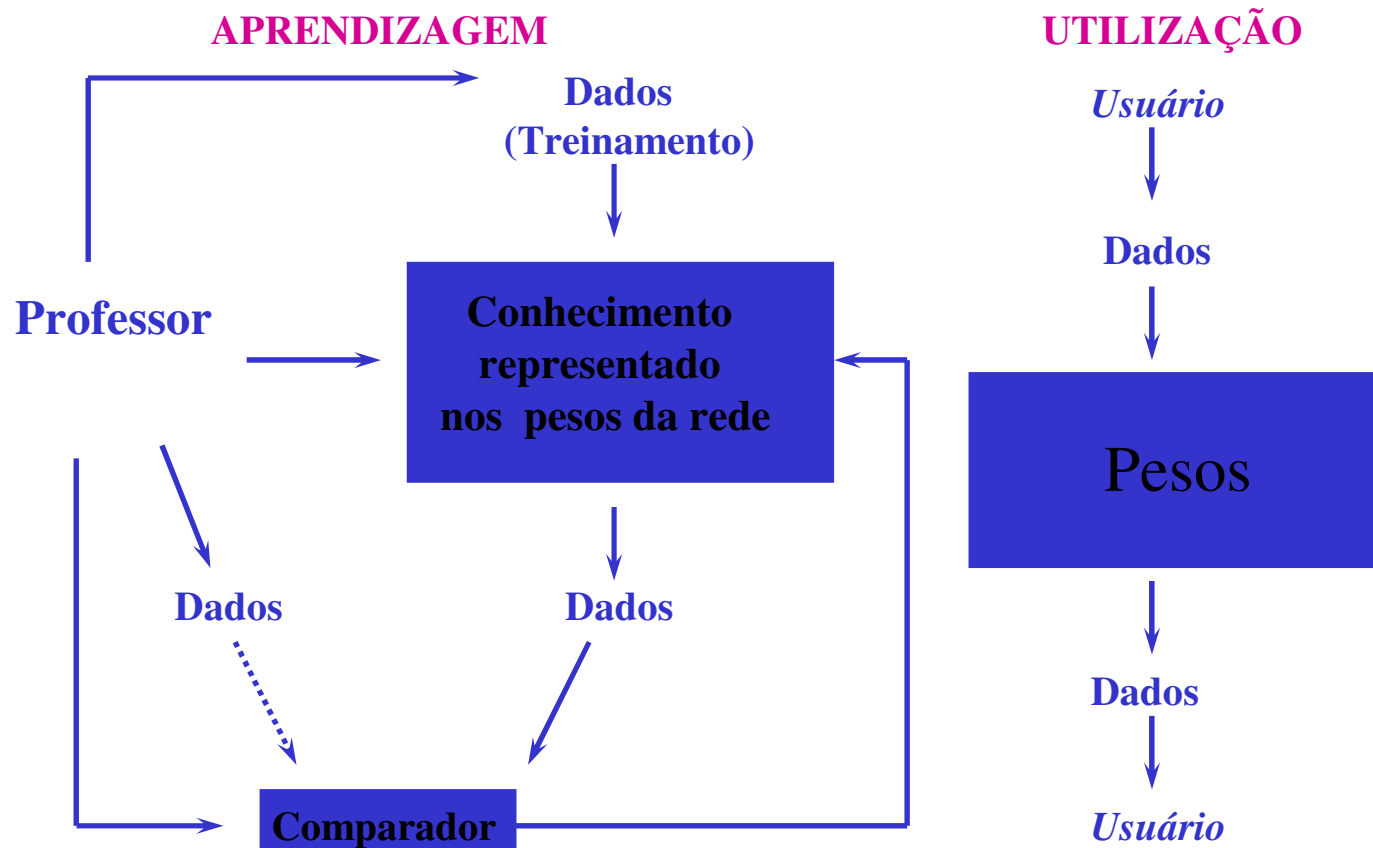
Retro-propagação dos erros

- Quando a saída gerada por uma rede neural não coincide com o estímulo de saída existe um erro que necessita ser corrigido.
- Redes de perceptrons multi-camada enfrentam este problema com a "**Atribuição de Créditos**", supondo que todos os EP e suas conexões devem partilhar a responsabilidade pelo erro
- A correção é feita propagando o erro para trás (para correção das conexões entre EP) pelas conexões da camada anterior até atingir a camada de entrada

Retro propagação

- Redes de Retro Propagação possuem uma camada de entrada, uma camada de saída e uma ou mais camadas intermediárias.
- Cada camada é completamente conectada à camada sucessora

Fases Aprendizagem/Utilização

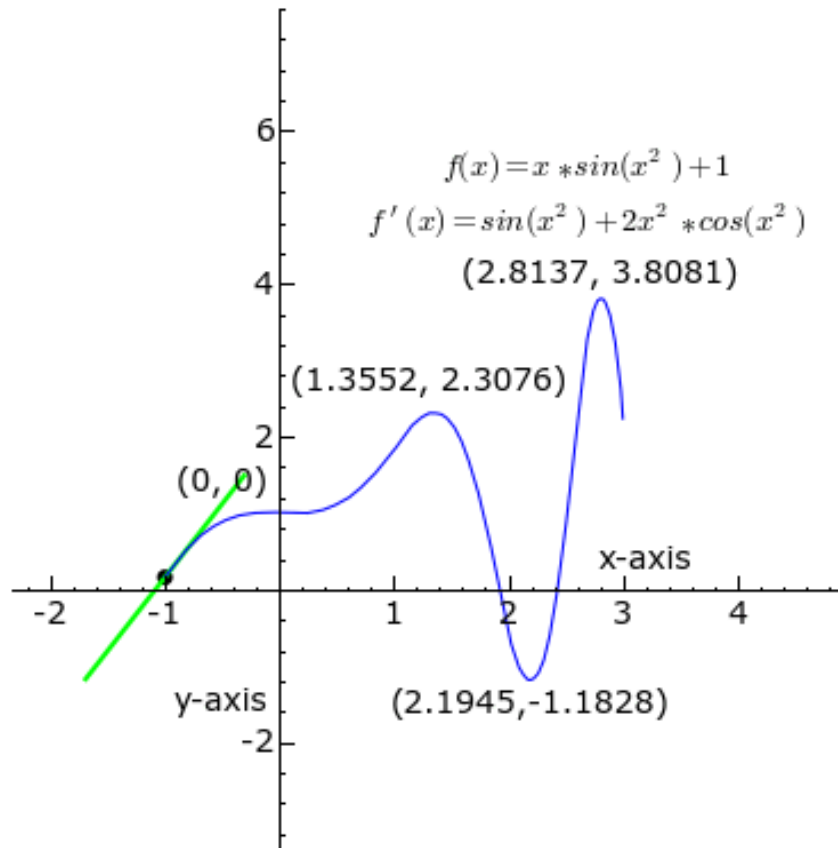


Mecanismo de Retro-propagação

- Propagar a entrada através das camadas ocultas até a saída
- Determinar o erro na camada de saída
- Propagar os erros de volta até a camada de entrada

A Retro Propagação não é utilizada na recuperação, apenas no treinamento

Derivada de uma função



O movimento no sentido de $-f'$ minimiza a função.

Lembrado a regra da cadeia

$$(f \circ g)'(x) = \left(f(g(x)) \right)' = f'(g(x))g'(x);$$

- Em notação de Leibniz:

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f(g(x))}{\partial g(x)} \cdot \frac{\partial g(x)}{\partial x}.$$

Lembrando derivadas totais

- Seja $f(t, x, y, z)$.
- A derivada total de f em t é:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial f}{\partial z} \frac{\partial z}{\partial t}$$

Princípios fundamentais

- Conjunto de dados para aprendizagem:

$$\Psi = \left\{ \langle x_1, y_1 \rangle, \dots, \langle x_p, y_p \rangle \dots \right\}.$$

- Erro ao propagar uma entrada:

$$J = \frac{1}{2} \sum_{j=1}^n (y_j - \hat{y}_j)^2$$

- Seguindo o princípio da minimização local:

$$\Delta w \propto -\frac{\partial J}{\partial w}.$$

Notação

- Rede multi-camada de L camadas ($l = 1 \dots L$).
- \hat{y}_j^l : saída do neurônio j da camada l .

$$\hat{y}_j^l = f(\text{net}_j^l) = f\left(\sum_i w_{ji}^l x_i\right)$$

- w_{ji}^l : peso i , do neurônio j da camada l .
- x_i^l : entrada i à camada l .
- Entradas da camada l são as saídas da camada $l-1$: $x_i^l = \hat{y}_i^{l-1}$.

Derivada do erro com relação ao peso w_{ji}^l :

$$\delta_j^l = \frac{\partial J}{\partial \text{net}_{ji}^l}$$

$$\frac{\partial J}{\partial w_{ji}^l} = \frac{\partial J}{\partial \text{net}_{ji}^l} \frac{\partial \text{net}_{ji}^l}{\partial w_{ji}^l}.$$

$$\frac{\partial \text{net}_{ji}^l}{\partial w_{ji}^l} = \frac{\partial (\sum_{i=1}^n w_{ji}^l x_i^l)}{\partial w_{ji}^l} = x_i^l$$

Podemos escrever: $\frac{\partial J}{\partial w_{ji}^l} = \delta_j^l x_i^l$

Deltas: camada de saída ($l = L$)

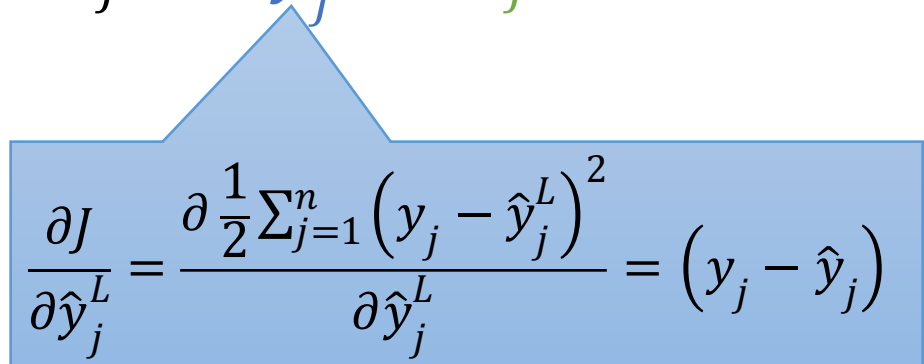
$$\frac{\partial J}{\partial \hat{y}_j} = \frac{\partial \frac{1}{2} \sum_{j=1}^n (y_j - \hat{y}_j)^2}{\partial \hat{y}_j} = (y_j - \hat{y}_j)$$

$$\delta_j^L = \frac{\partial J}{\partial net_j^L} = \frac{\partial J}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial net_j^L} = (y_j - \hat{y}_j) f'(net_j^L)$$

$$\frac{\partial \hat{y}_j}{\partial net_j^L} = f'(net_j^L)$$

Deltas: camada oculta ()

$$\delta_j^L = \frac{\partial J}{\partial net_j^L} = \frac{\partial J}{\partial \hat{y}_j^L} \frac{\partial \hat{y}_j}{\partial net_j^L} = (y_j - \hat{y}_j) f'(net_j^L)$$


$$\frac{\partial J}{\partial \hat{y}_j^L} = \frac{\partial \frac{1}{2} \sum_{j=1}^n (y_j - \hat{y}_j^L)^2}{\partial \hat{y}_j^L} = (y_j - \hat{y}_j)$$

Deltas: camadas ocultas

Considerando o erro J como uma função de todos os neurônios que recebem entrada do neurônio de interesse.

$$J = J(net_1^{l+1}, \dots, net_j^{l+1}, \dots)$$

Então, podemos dizer que:

$$\frac{\partial J}{\partial \hat{y}_j^l} = \frac{\partial J(net_1^{l+1}, \dots, net_k^{l+1}, \dots)}{\partial \hat{y}_j^l}$$

Camada oculta: derivadas totais

Aplicando as derivadas totais:

δ_k^{l+1} : deltas da camada posterior!

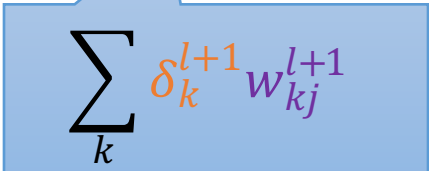
$$\frac{\partial J}{\partial \hat{y}_j^l} = \sum_k \left(\frac{\partial J}{\partial net_k^{l+1}} \frac{\partial net_k^{l+1}}{\partial \hat{y}_j^l} \right) = \sum_k \delta_k^{l+1} w_{kj}^{l+1}$$

$$\frac{\partial net_k^{l+1}}{\partial \hat{y}_j^l} = \frac{\partial \sum_j w_{kj}^{l+1} \hat{y}_j^l}{\partial \hat{y}_j^l} = w_{kj}^{l+1}$$

As entradas à
camada $l+1$ são as
saídas da camada l .

Retro-propagação do error às camadas ocultas

Os deltas das camadas superiores são usados para calcular os das camadas mais internas.

$$\delta_j^l = \frac{\partial J}{\partial net_j^l} = \frac{\partial J}{\partial \hat{y}_j^l} \frac{\partial \hat{y}_j^l}{\partial net_j^l} = \sum_k (\delta_k^{l+1} w_{kj}^{l+1}) f'(net_j^l)$$


A blue rectangular box highlights the summation term $\sum_k \delta_k^{l+1} w_{kj}^{l+1}$ from the equation above. A blue arrow points from the box to the $\delta \hat{y}_j^l$ term in the denominator of the second fraction in the equation.

Derivadas das funções de ativação

- Para a função logística $f(x) = \frac{1}{1 + e^{-x}}$

$$\frac{d}{dx} \frac{1}{1 + e^{-x}} = \frac{e^{-x}}{(1 + e^{-x})^2} = \left[\frac{-1 + 1 + e^{-x}}{1 + e^{-x}} \right] * \frac{1}{1 + e^{-x}} = \left[1 - \frac{1}{1 + e^x} \right]$$

$$\frac{\partial f(h_k)}{\partial h_k} = f'(h_k) = f(h_k) * (1 - f(h_k))$$

- Para a função tangente hiperbólica $f(y) = \tanh(y)$

$$\frac{d(\tanh(y))}{dy} = 1 - \tanh^2 y = \operatorname{sech}^2 y = \frac{1}{\cosh^2 y}$$

Resumindo

- Agora já sabemos como calcular os $\frac{\partial J}{\partial w}$ para todos os neurônios!
- Para os $\frac{\partial J}{\partial w}$ das camadas ocultas é preciso ter os deltas da camada anterior.
- Para cada entrada podemos calcular como variar os pesos:
- $\Delta w_{ji}^l = -\eta \frac{\partial J}{\partial w_{ji}^l} = -\eta \delta_j^l x_i^l.$

Algoritmo de retropropagação do erro

- Inicializar a rede com valores de pesos **aleatórios** e **pequenos**.
- Repetir hasta ***que seja suficiente***:
 1. Selecionar um par $\langle x_p, y_p \rangle \in \Psi$.
 2. Propagar x_p pela rede.
 3. Calcular os Δw_{ji}^l desde a camada de saída até a primeira camada oculta.
 4. Atualizar os pesos: $w_{ji}^l = w_{ji}^l + \Delta w_{ji}^l$.

Características do MLP

- Aproximador Universal de Funções
 - Uma única camada intermediária é capaz de aproximar qualquer função contínua definida em um hipercubo
- Alta capacidade de generalização
- Convergência para mínimo global não garantida
- Em alguns casos, lento na aprendizagem

Termo de “momentum”

- Caso uma porção do incremento de peso anteriormente calculado seja adicionado ao incremento atual pode-se modificar a equação, fazendo como que um filtro “passa-baixas” pelo qual as tendências gerais sejam reforçadas e o comportamento oscilatório seja inibido
- O termo de momentum, os tipos de aprendizagem Regra Delta Barra Delta e Delta Barra Delta Estendida são otimizações utilizadas para acelerar o treinamento em máquinas lentas e caíram em desuso

Termo de “momentum”

$$\Delta w_{ji}^{[s]} = -lcoef * e_j^{[s]} * x_i^{[s-1]} + momentum * \Delta w_{ji}^{[s-1]}$$

- Usualmente se utilizam como default
 - lcoef = 0,5
 - momentum = 0,9

Acumulação de pesos para a atualização

- Pode-se aumentar a velocidade de convergência fazendo a propagação de pesos depois do processamento de alguns pares de estímulos, em vez de o fazer logo após o processamento de cada par
- O número de pares de entrada e saída que é apresentado durante a acumulação é chamado de “época”

Redes de Retro Propagação

- Atualmente a arquitetura de Redes neuronais por retro propagação é a mais popular, eficaz e mais fácil de modelar para redes complexas e em múltiplas camadas
- Ela é mais usada que todas as outras arquiteturas juntas